Hybrid Systems, Lecture 5: Hybrid Automata

S. Nõmm

[1]Department of Computer Science, Tallinn University of Technology

17.03.2015

Hybrid Automaton

- A hybrid automaton is a formal model of a hybrid system.
- A hybrid automaton is a transition system that is extended with continuous dynamics.
- Formally: A hybrid Automaton is a tuple $H = (Q, V, Init, f, D, G, R, E$, where
  - $Q = q_1, \ldots, q_k$ - is a finite set of discrete states (control locations);
  - $X = (x_1, \ldots x_n)$ - is a finite set of continuous variables;
  - $f : Q \times \mathbb{R}^n \to \mathbb{R}^n$ - is an activity function;
  - $Init \subset Q \times \mathbb{R}^n$ - is the set of initial states;
  - $D : Q \to 2\mathbb{R}^n$ - invariants of the locations (domains);
  - $E \subseteq Q \times Q$ - is the transition relation;
  - $G : E \to 2^{\mathbb{R}^n}$ - is is the guard condition;
  - $R : E \to 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n}$ - is the reset map;

  This definition does not take into account synchronization labels.

Read and explain why.

Solution of Hybrid Automaton

A solution $\mathcal{X} = (\tau, q, x)$ of the hybrid automaton H consists of

- *Time trajectory* τ - a time line where solution is defined
- *State trajectory* (q, x) -state evolution of the hybrid automaton defined on τ

Time trajectory τ

A sequence of time intervals

$$\tau = \{I_i\}_{i=0}^{N}$$

such that

- $I_i = [\tau_i, \tau_i'] \quad \forall i < N$ where $\tau_i \leq \tau_i' = \tau_{i+1}$;
- If $N < \infty \Rightarrow \big(I_N = [\tau_N, \tau_N']\big) \bigwedge \big(I_N = [\tau_N, \tau_N')\big)$

Explain last phrase.

Solution $\mathcal{X} = (\tau, q, x)$

$$\tau = \{I_i\}_{i=0}^{N}, \quad q : \langle\tau\rangle \to Q, \quad x = \{x^i : i \in \langle\tau\rangle\}, \quad x^i : I_i \to X$$

such that:

- *Initialization* $(q(0), x^0(0)) \in \mathrm{Init}$;
- *Time driven* $\forall t \in [\tau_i, \tau_i'), \quad \dot{x}^i(t) = f(q(i), x^i(t))$ and $x^i(t) \in D(q(i))$
- *Event driven* $\forall i \in \langle\tau\rangle \setminus N, e = (q(i), q(i+1)) \in E$, $x^i(\tau_i') \in G(e)$ and $x^{i+1}(\tau_{i+1}) \in R(e, x^i(\tau_i'))$

Transition Relation for Hybrid Automaton

- To each discrete state $q \in Q$ associate a differential equation $\dot{x} = f_q(x)$.
- To each generator g linked to an edge $e \in Q \times Q$ associate a guard $G : Q \times Q \to 2^{\mathbb{R}^n}$
- Transition relation consists of two parts:
  - *Time driven* $(q, x) \to (q, y)$, *Should something be added here?*
  - *Event driven* $(q, x) \to (q, y')$, *Should something be added here?*

Properties of Hybrid Automata

- *Liveness* For all $(q_0, x_0) \in Init$, there exists at least one (infinite) solution from (q_0, x_0);
- *Determinism* For all $(q_0, x_0) \in Init$, there exists at most one solution starting from (q_0, x_0);
- *Zenoness* Finite execution time

$$\tau_\infty = \sum_{i=1}^{\infty} (\tau_i - \tau_i') < \infty;$$

- *Stability of equilibria* and other invariant sets;
- *Reachability* states $Reach \in Q \times X$;

Zeno of Elea (490 – 430 B.C.)

- Born in southern Italy
- Met Socrates in Athens 449 B.C.
- Went back to Elea and into politics
- Tortured to death
- Paradoxes proved that motion and time are illusions
- Led to mathematical problems not solved until 19th century

Zeno

- A solution is Zeno if it exhibits infinitely many discrete jumps in finite time.
- Zeno is a truly hybrid phenomenon: it cannot be formulated for a purely discrete system without the notion of continuous time.
- Zeno is due to that the model does not reflect reality with sufficient detail.
- A hybrid automaton has Zeno solutions only if (Q, E) is a cyclic graph.
- The convergence point of a Zeno solution is denoted *Zeno state*.
- Zeno states lie on the intersection of guards.

Switched systems

Let $\Omega_q, q = 1, \ldots, m$ denote a partition of the continuous state space $\mathbb{R}^n$.
A *switched system* is then defined as

$$\dot{x} = f_q(x), \quad x \in \Omega_q$$

Switched System as Hybrid Automaton

- $$\dot{x} = f_q(x), \quad x \in \Omega_q$$
  corresponds to the hybrid automaton
- $Q = \{1, \ldots, m\}$, $X = \mathbb{R}^n$, $Init \in \{q\} \times \Omega_q$
- $f(q, x) = f_q(x)$;
- $D(q) = \Omega_q$;
- $(q, q') \in E$ if $D(q)$ to $D(q')$ such that $D\overline{(q)} \cup D\overline{(q')} \neq \emptyset$
- $G(q, q') = D\overline{(q)} \cup D\overline{(q')}$
- $R(q, q', x) = x$

Example 1: Bouncing ball

- Free fall: $\ddot{y} = -g$
- Collision:

$$\begin{aligned} y^+(t) &= y^-(t) = 0 \\ \dot{y}^+(t) &= -c\dot{y}^-(t) \end{aligned}$$

$c \in [0, 1]$ energy reflected at impact. item

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -g \end{aligned}$$

Example 2: Thermostat

Goal: is to keep temperature about 22 degrees.

- $\dot{x} = -x + 18$
- $\dot{x} = -x + 25$
- Event based control

Tank system

Goal is to prevent the tank from emptying or filling up.

- λ - pump-on inflow, μ - constant outflow, δ - delay between sending and executing pump command;
- Pump off $\dot{y} = -mu$
- Wait to On

$$\begin{aligned} \dot{y} &= -\mu \\ \dot{\tau} &= 1 \end{aligned}$$

- Pump On $\dot{y} = \lambda - \mu$
- Wait to Off

$$\begin{aligned} \dot{y} &= \lambda - \mu \\ \dot{\tau} &= 1 \end{aligned}$$

Example 3:Switched server

Parts are incoming through the n buffers.

- $\delta_{i,j}$ - is the set up time required to move from budder i to j
- Algorithm:
  - Start with budder 1
  - Work on a buffer until empty
  - when budder j is empty move to the buffer $j + 3 \mid n$

Server system with congestion control

- r -incoming rate, q_{max} - is the maximum of element allowed
  B- bandwidth, rate of service
- Additive/multiplicateive increase of r while $q < q_{max}$
- $q \geq q_{max}$ multiply by $\gamma \in (0, 1)$
- $\dot{q} = r - B$ and $\dot{r} = 1$